CLAIMS

I claim:

1. A cache coherency system for a shared memory parallel processing system including plurality of processing nodes, comprising: a multi-stage communication network for interconnecting said processing nodes; each said processing node including one or more caches for storing a plurality of cache lines; and a cache coherency directory which is distributed to each of said nodes for tracking which of said nodes have copies of each cache line.

2. A shared memory parallel processing system including a plurality of processing nodes, comprising:

a multi-stage communication network for interconnecting said processing nodes, said network including a plurality of self-routing switches cascaded into first, middle and last stages, each said switch including a plurality of switch inputs and a plurality of switch outputs, each of said switch outputs of each said switch coupled to a different switch input of others of said switches, switch outputs of said last stage switches including network output ports, and switch inputs of said first stage switches comprising network input ports;

each processing node including:

a network adapter for transmitting and receiving messages with respect to other processing nodes over said network;

a local processor;

at least one private write-through cache;

a section of shared memory organized into a plurality of cache lines, each cache line including one or more addressable memory locations;

a cache coherency directory for tracking which of said nodes have copies of each cache line;

said local processor at a first processing node being operable for writing data to said private cache at said first node, as the same data is written to either shared memory at said first node or sent over said network for writing to the shared memory and private cache of a second processing node.

3. The shared memory parallel processing system of claim 2, wherein said section of shared memory is divided into first and second portions, said first portion for storing unchangeable data, and said second portion for storing changeable data.

4. The shared memory parallel processing system of claim 3, said cache coherency directory for this processing node listing which nodes of the plurality of nodes have accessed copies of said cache lines in said second portion of shared memory at this processing node.

5. The shared memory parallel processing system of claim 4, wherein each said processing node is operable for reading, storing, and invalidating the shared memory at any of said plurality of processing nodes selectively by transmitting and receiving messages over said network, a first message type for requesting the read of a cache line, a second message type for returning the requested cache line, a third message type for storing a cache line, and a fourth message type for invalidating a cache line.

6. The shared memory parallel processing system of claim 5, said network adapter further comprising:

a first buffer for transmitting to said network shared memory read command messages of said first message type and said second message type;

a second buffer for transmitting to said network shared memory store command messages of said third message type;

a third buffer for transmitting to said network invalidate messages for said cache coherency directory of said fourth message type;

a fourth buffer for receiving from said network shared memory read command messages of said first message type and said second message type;

a fifth buffer for receiving from said network shared memory store command messages of said third message type; and

a sixth buffer for receiving from said network invalidate messages for said cache coherency directory of said fourth message type.

7. A shared memory parallel processing system, comprising:

a plurality of nodes, each node including a node memory, at least one cache, and a memory controller;

a multi-stage switching network for interconnecting said processing nodes, said switching network including a plurality of self-routing switches cascaded into first, middle and last stages, each said switch including a plurality of switch inputs and a plurality of switch outputs, each of said switch outputs of each said switch coupled to a different switch input of others of said switches, switch outputs of said last stage switches including network output ports, and switch inputs of said first stage switches comprising network input ports;

a system memory distributed to said node memories of said plurality of nodes and accessible by any node; each said node memory being organized into a plurality of addressable word locations;

said memory controller at this node operable for performing local memory access to the portion of system memory at this node and for performing remote memory access over said network to the portion of system memory at other nodes; and

a cache coherency controller at this node being responsive to both local memory accesses and remote memory accesses to data stored in a word location of said node memory at this node for caching accessed data in the cache of this node and for communicating data for assuring cache coherency throughout said system over said network.

8. The shared memory processing system of claim 7, said system memory being distributed in equal portions to each said node memory; and said node memory being further sub-divided into a first memory section for storing data that is changeable and a second memory section for storing data that is unchangeable.

9. The shared memory processing system of claim 7, further comprising node indicia for uniquely identifying each node.

10. The shared memory processing system of claim 7, said cache coherency controller further comprising:

an invalidation directory for storing a list of node indicia identifying those nodes having accessed a copy of each said cache line of node memory since the last time the cache line was changed.

11. The shared memory processing system of claim 10, said cache coherency controller further comprising:

an overflow directory for expanding said invalidation directory when the list of node indicia for a cache line becomes too long to contain entirely with said invalidation directory.

12. A shared memory parallel processing system, comprising:

a plurality of nodes, each node including a node memory, at least one cache, and a memory controller;

a multi-stage switching network for interconnecting said processing nodes, said switching network including a plurality of self-routing switches cascaded into first, middle and last stages, each said switch including a plurality of switch inputs and a plurality of switch outputs, each of said switch outputs of each said switch coupled to a different switch input of others of said switches, switch outputs of said last stage switches including network output ports, and switch inputs of said first stage switches comprising network input ports; and

a network adapter responsive to a node connection request for establishing a connection path to a target node, first by attempting to establish a quick connection path across a plurality of segments of said switching network to said target node, and upon determining any one of said plurality of segments is not available, issuing a camp-on connection request to said target node.

13. The shared memory parallel processing system of claim 12, further comprising:

said plurality of nodes each coupled to one of the network output ports and to one of the network input ports;

each node further including:

receive means for receiving a data message; and

send means for sending a data message across an n-stage switching network from a local node to a remote node, said send means generating said connection request including n sequential connection commands, each sequential connection command selecting one of said plurality of connection segments for each of the n switch stages of said network.

14. The shared memory parallel processing system of claim 12, each said switch being responsive to node connection requests and camp-on connection requests for establishing connection segments from any switch input port to any switch output ports.

15. The shared memory parallel processing system of claim 14, each said switch further comprising:

a data bus for transferring said data message;

a rejection control line for signalling back to a sending node a rejection of any connection request;

an acceptance control line for signalling back to said sending node the acceptance of a camp-on connection request;

a valid control line for receiving from said sending node the activation of a node connection request; and

a camp-on control line for receiving from said sending node the activation of a camp-on connection request.

16. A bi-directional network adapter for interfacing a local node of a shared memory parallel processing system to a multi-stage switching network for communications with a remote node, each said node including a node memory including a changeable portion and an unchangeable portion, and a node cache; said network adapter comprising:

a plurality of send buffers for storing and forwarding data messages from said local node to said remote node over said network, and

a plurality of receive buffers for storing and forwarding a plurality of data messages from said remote node to said local node over said multi-stage network;

said data messages including:

an invalidation message for invalidating a cache line that was accessed by a remote node after said cache line has changed;

a read request message for requesting access of a cache line from a remote node;

a response message for returning a cache line over the network to a remote node that has previously requested data by a read request message; and

a store message storing a changed cache line to a remote node.

17. The network adapter of claim 16, said data messages further including a message header comprising:

message type differentiation indicia;

destination node indicia for identifying a node for receiving said data message over said network;

source node indicia for identifying a node for transmitting said data message over said network;

message length indicia for defining the variable number of words included in said data message;

memory area indicia for defining whether memory words included in said data message are read from said changeable area;

time indicia for defining the time of generation of said data message; and

memory address indicia for defining the address location in memory of the memory word included in said data message.

18. The network adapter of claim 17, said send buffers further comprising:

a read send FIFO for storing and forwarding read request messages and response messages from said local node to said remote node;

a store send FIFO for storing and forwarding store messages from said local node to said remote node; and

an invalidation send FIFO for storing and forwarding invalidation messages from said local node to said remote node;

and said receive buffers further comprising:

a read receive FIFO for storing and forwarding read request messages and response messages from said remote node to said local node;

a store receive FIFO for storing and forwarding store messages from said remote node to said local node; and

an invalidation receive FIFO for storing and forwarding invalidation messages from said remote node to said local node.

19. The network adapter of claim 18, further comprising:

a send FIFO selection means for prioritizing the selection of a data message from one of said three send FIFO means for transmission to said network by first selecting data messages from said invalidation send FIFO and thereafter alternatively selecting data messages from said read and store send FIFOs;

a receive FIFO selection means responsive to said message type differentiation indicia for selecting one of said three receive FIFO means for storing a data message received from said network; and

said network adapter being responsive to a node connection request for establishing a connection path to a target node, first by attempting to establish a quick connection path across a plurality of segments of said switching network to said target node, and upon determining any one of said plurality of segments is not available, issuing a camp-on connection request to said target node.

20. A memory controller for a local node of a shared memory parallel processing system, said node including a node processor, a node memory, a node cache and an I/O adapter, said system including a multi-stage switching network for communications amongst said local node and a plurality of remote nodes, said node memory including a changeable portion and an unchangeable portion; said memory controller comprising:

first means responsive to a request by said processor for access to a memory word for first accessing said node cache of said local node; and

second means responsive to said first means being unable to access said memory word in said node cache for accessing said memory word selectively from a cache line in said node memory or remote memory and storing said cache line to said node cache.

21. The memory controller of claim 20, further comprising:

remote fetch interrupt means for issuing an interrupt signal to said node processor upon determining that a requested memory word is located in remote memory for causing said node processor to switch from a first instruction stream thread to a second instruction stream thread.

22. The memory controller of claim 20, further comprising:

data message generation means responsive to a request from a remote node for accessing a cache line identified by a remote request read address for generating a read response message to return the accessed cache line to said remote node, said read response message including a message header comprising

message differentiation indicia for defining said read request message type;

destination node indicia equal to the sector segment of said node memory for said addressed memory word;

source node indicia set to the node ID number of the local node;

message length indicia for defining said read request message as being comprised of said message header only; and

memory address indicia for specifying the memory address of said memory word;

said data message generation means further operable for delivering said read response message to a read send FIFO of said network adapter for transmission to said network and the remote node selected by said destination node indicia.

23. The memory controller of claim 20, further comprising:

an invalidation directory;

cast-out means for deleting a cache line from said node cache when said cache is full to provide space for a new cache line to be stored to said cache; and for sending the address of the deleted cache line to said invalidation directory to indicate said node no longer has a copy of said cache line.

24. The memory controller of claim 23, further comprising:

cast-out message generation means responsive to said cast-out means deleting a cache line addressed to a remote node for generating a cast-out message to said remote node to send the cast-out address and the local node ID number to said remote node over said network;

cast-out message receiving means for delivering a cast-out address and the source node ID number from the message header of a cast-out message to said invalidation directory.

25. The memory controller of claim 20, further comprising:

cache copy update means for sending cache update messages to update corresponding cache lines all remote nodes having copies of a changed cache line; and

cache update message receiving means for replacing a cache line of data with an updated cache line of data received from a remote node.

26. The bi-directional network adapter of claim 16, said data messages further comprising:

a cast-out message for invalidating an invalidation directory entry at a remote node for this local node;

a cache copy update message for updating copies of a changed cache line at this local node at remote nodes having copies of said changed cache line; and

a node indicia assignment message for sending a different node number to each of the plurality of nodes of the system.

27. A method for operating memory controller for a local node of a shared memory parallel processing system, said node including a node processor, a node memory, a node cache and an I/O adapter, said system including a multi-stage switching network for communications amongst said local node and a plurality of remote nodes, said node memory including a changeable portion and an unchangeable portion; the method comprising the steps of:

responsive to a request by said processor for access to a memory word, accessing said node cache of said local node; and thereafter

responsive to said first means being unable to access said memory word in said node cache, accessing said memory word selectively from a cache line in said node memory or remote memory and storing said cache line to said node cache.

28. The method of claim 27, further comprising the step of:

issuing an interrupt signal to said node processor upon determining that a requested memory word is located in remote memory for causing said node processor to switch from a first instruction stream thread to a second instruction stream thread.

29. A method for operating bi-directional network adapter for interfacing a local node of a shared memory parallel processing system to a multi-stage switching network for communications with a remote node, each said node including a node memory including a changeable portion and an unchangeable portion, and a node cache; comprising the steps of:

operating a plurality of send buffers for storing and forwarding data messages from said local node to said remote node over said network, and

operating a plurality of receive buffers for storing and forwarding a plurality of data messages from said remote node to said local node over said multi-stage network;

said data messages including:

an invalidation message for invalidating a cache line that was accessed by a remote node after said cache line has changed;

a read request message for requesting access of a cache line from a remote node;

a response message for returning a cache line over the network to a remote node that has previously requested data by a read request message; and

a store message storing a changed cache line to a remote node.

30. The method of claim 29, further comprising the steps of:

operating a read send FIFO for storing and forwarding read request messages and response messages from said local node to said remote node;

operating a store send FIFO for storing and forwarding store messages from said local node to said remote node; and

operating an invalidation send FIFO for storing and forwarding invalidation messages from said local node to said remote node;

operating a read receive FIFO for storing and forwarding read request messages and response messages from said remote node to said local node;

operating a store receive FIFO for storing and forwarding store messages from said remote node to said local node; and

operating an invalidation receive FIFO for storing and forwarding invalidation messages from said remote node to said local node.